2018 IEEE 12th International Conference on Semantic Computing

Adjective Intensity Ordering by Representing Word Definitions as a System of Linear Equations

Stefan Eng
CRESST, UCLA
Los Angeles, USA
Email: seng@cresst.org

Jennifer Tan
UCLA
Los Angeles, USA
Email: jen213@ucla.edu

Markus Iseli
CRESST, UCLA
Los Angeles, USA
Email: iseli@cresst.org

***Abstract*—While speakers may disagree or have difficulty deciding the ordering of semantically similar adjectives such as icy and frosty, arranging adjectives on a linear scale is an intuitive task for most speakers of any language. Currently, there is no standard method or database to order gradable adjectives that share a common attribute, such as temperature or speed. Modeling how adjectives relate to one another semantically is an important aspect of language understanding. Applications of adjective intensity ordering include detecting scale differences in reviews and measuring similarity of larger segments of text. This paper discusses a novel approach that solves a system of linear equations constructed only from word dictionary definitions to compare the intensity of similar adjectives. We used Krippendorff's alpha to measure the agreement between a human standard and our algorithms' results on an ordinal scale and attained a score of 0.927 for temperature-, 0.652 for speed-, and 0.673 for happiness-related adjectives. As a reference, Krippendorff inter-rater agreement was 0.927 for temperature, 0.816 for speed-related adjectives, and 0.849 for happiness-related adjectives.**

I. Introduction

Current word embedding approaches, like GloVe [1], train word vectors on co-occurrence metrics but do not provide a reliable ordering of words along attribute dimensions within clusters, such as temperature, speed, or happiness, which are all determined by the meaning of adjectives.

WordNet [2] is currently the most up-to-date and comprehensive corpus of English nouns, verbs, adjectives, and adverbs. While this collection contains useful word relations including antonymy, synonymy, hyponymy, meronymy, and entailment, it does not provide a metric to distinguish different degrees of adjective similarity. The similarity feature in WordNet groups all similar adjectives into one group without adequately capturing or conveying the notion that some adjectives are more closely related than others. For instance, WordNet categorizes both arctic and chilly as similar words to cold, although most people would easily say that arctic is far more cold than chilly.

Previous work related to word or adjective similarity relies on large amounts of data or does not succinctly represent the notion of similarity. Sheinman and Tokunaga [3] [4] present a way to order similar adjectives by their intensities, which they named AdjScales. This approach involves finding patterns from the Web that indicate a stronger-than relationship between two words, such as great and good. AdjScales is able to order adjectives using these Web-extracted patterns in conjunction with WordNet and they report a pairwise score between annotators of 57.87% precision and 59.48% recall on 298 adjective sets, which is promising considering that their human rater score was 63.49% for precision and 63.49% for recall. They acknowledge limits to their approach, such as how AdjScales does not have a neutral relation. For instance, most native English speakers would classify the two speed adjectives, quick and fast, as having the same strength. AdjScales can only classify similar words as stronger (more intense), weaker (milder), and unconfirmed. They also acknowledge issues with scaling, such as when a word does not occur in any of their patterns.

Wang et al. [5] expanded adjectives and calculated the sentiment intensity of words using a method they called POlarity Adjective Expanding (POAE). Their approach makes use of the synonym, antonym, similar-to, and see-also relations in WordNet and does not make use of word definitions. In this paper, we explore how semantic analysis of word definitions can uncover adjective ordering information on a linear scale by representing the semantic relationships as a system of linear equations. We used a novel approach of only using word definitions extracted from online dictionaries and do not use other text resources like the Web or Wikipedia. Section 2 describes our methods and data, Section 3 shows the results, and Section 4 contains the conclusion and discusses limits of our method as well as areas for future work.

II. Methods and Data

Our approach of using word definitions and a system of linear equations to perform adjective intensity ordering, consists of five steps, outlined in the subsections below.

A. Clustering of Attribute Adjectives

To perform intensity ordering of adjectives, we decided to focus on adjective clusters that have the objective physical attributes $temperature$ and $speed$ and the subjective attribute $happiness$. All of these adjectives are gradable adjectives, which can be compared against each other and scaled in degree [6]. We then extracted the adjectives for each cluster from WordNet using the relations similar-to, has-attribute, see-also, and has-lemma. Note that the WordNet relation has-lemma does not refer to the usual sense of lemma as the base form of a word, but is comparable to the similar-to relation. The words from the has-attribute relation are $cold$, $cool$, $warm$, and hot for temperature; $fast$ and $slow$ for speed; and $happy$ and $unhappy$ for happiness.

0-7695-6360-0/18/$31.00 ©2018 IEEE
DOI 10.1109/ICSC.2018.00047

IEEE computer society

To automatically expand the adjective clusters, we included adjectives using the similar-to relation on the adjectives we retrieved using the has-attribute relation and included their WordNet lemmas. We removed adjectives that were archaic (belonging to the archaism usage_domain in WordNet), had no Wiktionary definition, or did not describe the attribute itself, such as $fiery$ and $igneous$, which are related or suggestive of fire but do not describe temperature, or *air-conditioned* and *air-cooled*, which are describing the method of temperature change, or $precooled$, which describes the time of temperature change. The final count of extracted adjectives was 46 for temperature, 19 for speed, and 174 for happiness, see Tables II and III.

B. Retrieval of Adjective Definitions

For all adjectives we collected definitions from the online dictionaries Wiktionary, Oxford, and WordNet. This step involved finding the definition of the correct word sense. For all dictionaries, we selected the first word sense definition that contained a word from our list of keywords, which consisted of the attribute (i.e. temperature, speed, or happiness) and adjectives in the current adjective cluster. If none of the definitions contained a keyword, we selected the first definition. Note that the order of word senses in each dictionary is commonly determined by their probability of occurrence, highest probably first. If none of the definitions contained a keyword, we tried to use synonyms from PorterStemmer to find matches in the stems of words rather than looking for an exact match. While using three definition sources and PorterStemmer helped to connect most of the definitions, there were certain words we could not include. We had to exclude the following adjectives because of missing keywords: $frosty$, $rimed$, $rimy$, for the attribute temperature; and $heartbroken$, $lovesick$, $mirthless$, etc., for the attribute happiness.

C. Construction of System of Linear Equations

The goal of this step was to assign adjectives with scalars that indicate adjective intensity on a linear scale. These scalars can range from negative to positive infinity, but are limited in our case by what we assign to *high_val*, a constant acting as an anchor that defines the strength value of the high attribute (*high_prop*), which represents represents either high temperature, high speed, or high happiness. For example, using the Wiktionary definitions for hot as *having a high temperature*, and for cold as *having a low temperature*, we create the following equations:

$hot = high_prop$
$cold = low_prop$
$low_prop = -high_prop$
$high_prop = high_val.$

In all our calculations, we anchored the high value (*high_val*) at 10 but could have chosen any other positive value. Adjective definitions from all three dictionaries were automatically converted into a system of m linear equations with n unknowns, $Ax = b$, where A is an $m \times n$ matrix, x is an $n \times 1$ column vector, and b is an $m \times 1$ column vector. Adjective intensities are the unknown variables represented in vector x, adverbs, such as extremely or fairly, and strength-related adjectives, such as low or high, are multiplicative constants, which are represented in matrix A, and vector b is zero with the exception the element corresponding to the index of *high_prop*. We used spaCy (https://spacy.io) to find dependency relations in the word definitions to determine which adjectives were modified by adverbs and which attribute instances in the definitions were modified by adjectives indicating positivity or negativity.

TABLE I. INTENSIFIERS AND DOWNTONERS

Score	Adverbs
1.4	extremely, intensely, exceptionally, astoundingly, excessively, totally, absolutely, completely, oppressively, incredibly, remarkably, particularly, unusually, amazingly, unbearably, utterly, dangerously, extraordinarily, really, very, highly
0.8	fairly, pretty, quite, rather, moderately, somewhat
0.6	reasonably, slightly, a little, mildly, a bit, pleasantly
-0.6	not quite

The multiplicative constants were assigned as follows. For strength-related adjectives that modified the cluster attribute we assigned a value of 1 to high, good, great, higher, better and value of -1 to low, bad, little, lower, and worse. As an illustrative example, the definition for $cold$ contains *low temperature*, where temperature is the attribute of the temperature adjective cluster, thus *low* is assigned a value of -1. For adverbs, the multiplicative constants are expressed as intensity scores and are listed in Table I. *Intensifiers*, such as extremely, increase the intensity of adjectives and were assigned scores greater than 1. *Downtoners*, such as slightly, downplay the intensity of adjectives, and were given positive scores less than 1. There exist approaches to automatize the process of adverb intensity scoring such as described by [7], which we did not explore due to our focus on adjective intensity ordering rather than adverb intensity scoring.

Given the keywords and the multiplicative constants, the retrieved definitions were then split into shorter definitions and converted to equations. For example, the WordNet definition for $cool$ is *Having a slightly low temperature; mildly or pleasantly cold*. We split this into the two definitions *having a slightly low temperature* and *mildly or pleasantly cold*. Using the adverb intensity scores from Table I, the values of slightly and pleasantly in the definitions are set to 0.6. The former definition then becomes the equation
$cool = 0.6 * low\ temperature$
and the latter definition becomes the equation
$cool = 0.6 * cold.$
Rearranging the equations and putting everything together for our three adjective examples (hot, cold, cool), we get the following equations:

$hot - high_prop = 0$
$cold + high_prop = 0$
$cool - 0.6 * low_prop = 0$
$cool - 0.6 * cold = 0$
$low_prop + high_prop = 0$
$high_prop = 10.$

Thus, for this example,
$m = 6$, $n = 5$, $x = [cold, cool, hot, low_prop, high_prop]^T$,
matrix A becomes

$$\begin{bmatrix} 0 & 0 & 1 & 0 & -1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & -0.6 & 0 \\ -0.6 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and vector $b = [0, 0, 0, 0, 0, 10]^T$.

To ensure that our system of linear equations was connected such that each equation contained at least one variable (keyword) that was contained in at least one other equation, in addition to using definitions from multiple dictionaries, we allowed matches between stemmed and unstemmed words using PorterStemmer, for example $tropical$ to match $tropic$, and matches between attribute (noun) synonyms. However, the stemming and attribute synonyms did not prove to be much help, as the only pair of words we connected using the PorterStemmer for temperature was $tropical$ to match $tropic$, and we found no helpful attribute synonyms for temperature. Additionally, to increase equation connectedness, we updated the method to check for synonyms of the attribute in the word definitions using PyDictionary [8].

D. Solve System of Linear Equations

Because there exist usually more than one definition per adjective, the system of linear equations is over-determined, meaning that the number of equations is larger than the number of unknowns. Henceforth, to solve our system of linear equations, we used the least squares approach. Specifically, we used the Python algorithm numpy.linalg.lstsq, which solves the equation $Ax = b$ by computing a vector x that minimizes the Euclidean 2-norm $||b - Ax||^2$.

After solving for x, the adjectives were sorted linearly in the order of increasing values. An adjective with a higher absolute value corresponded to a greater intensity, positive values indicated a presence of the attribute, negative values indicated a lack of the attribute, and adjectives with the same value meant they were equal in intensity levels.

E. Intensity ordering by human raters

Four fluent English speakers set our standard for the temperature and speed adjective intensity ordering. There were three raters for the happiness adjectives. The raters unanimously agreed to remove some of the adjectives in the adjective attribute clusters that seemed too uncommonly used or incomparable on the same scale as the rest of the adjectives. For instance, words such as water-cooled and calorific were omitted from the temperature rankings. Raters were given a list of adjectives and their definitions extracted as described in section 2.1 and 2.2 above, and then asked to rank order them. Raters were instructed to give the same scores to adjectives that appeared to be of the same intensity and to keep increasing the count. For example, instead of a ranking like [1,2,3,3,3,3,4,...], where the same score of three is given to four adjectives, they needed to keep counting to get, correctly, [1,2,3,3,3,3,7,...]. The orderings were combined into one standard for each attribute using Schulze's method [9], a voting algorithm. Each rater is a voter, that votes for a candidate (the word) based on the adjective intensity scale. The sorted list of winners becomes the standard with which we compare the linear system results.

TABLE II. TEMPERATURE AND SPEED

Temperature	**LS**	**H**	**Speed**	**LS**	**H**
arctic	1	1	bumper-to-bumper	1	1
stone-cold	2	7	laggard	2	3
gelid	3	5	poky	3	5
glacial	4	2	slow	4	4
icy	5	3	sluggish	5	2
ice-cold	6	4	high-speed	6	17
bleak	7	11	rapid	7	14
chilly	8	12	windy	8	7
cold	9	9	quick	9	11
frigorific	10	15	smart	10	10
parky	11	13	alacritous	11	13
frosty	12	8	fast	12	12
algid	13	10	pokey	13	6
crisp	14	16	speedy	14	9
nippy	15	14	meteoric	15	19
cool	16	17	fleet	16	16
frigid	17	6	swift	17	15
tepid	18	22	blistering	18	18
hottish	19	26	winged	19	8
lukewarm	20	21			
warm	21	23			
unwarmed	22	20			
heated	23	25			
air-conditioned	24	18			
heatless	25	19			
hot	26	28			
red-hot	27	39			
stifling	28	32			
sultry	29	31			
thermal	30	27			
tropic	31	30			
tropical	32	29			
warmed	33	24			
blistering	34	36			
sizzling	35	37			
baking	36	35			
scorching	37	38			
overheated	38	34			
torrid	39	33			

III. RESULTS AND DISCUSSION

We evaluated our algorithm on temperature data and then tested on speed and on happiness, without changing our algorithm.

Table II shows the adjective intensity ordering results of our algorithm (LS) compared to the human scored results (H). For temperature, 1 is the coldest and 39 is the warmest. For speed 1 is the slowest, and 19 is the fastest. Adjective intensity order for the three attributes: temperature and speed. Happiness adjectives are included in the appendix in Table III. Adjectives are sorted by algorithm-rated intensity scores (LS), the automatic scores generated with our algorithm, which are compared with the human standard rating (H).

Out of the original clusters, some of the words and definitions could not be used since they had no link to other definitions. This reduced the list of adjectives from 46 to 39 for temperature, from 174 to 137 for happiness and all 19 definitions were used for speed. For temperature, the four raters orderings on 46 adjectives pertaining to temperature had a Krippendorff's alpha [10] of 0.957 with ordinal distance as the measure. The table on the last page compares the raters ordering with the linear systems ranking. We used Krippendorff's alpha, calculated with the R package irr [11], to measure the similarity between a human standard and our algorithms results with ordinal distance as the measure and

attained a score of 0.927 for temperature, 0.652 for speed and 0.673 for happiness. Inter-rater agreement was 0.957 for temperature, 0.816 for speed, 0.849 for happiness.

IV. Conclusion and Future Work

Adjective intensity ordering using definitions and a system of linear equations is a new method of language modeling we explored in this paper. A limitation of our approach is that the semi-automatic attribute adjective clustering is based on WordNet, which is a manually-created database. The success of our intensity rankings is dependent on the dictionary entries and the definitions that we are able to retrieve. Future work could explore word-sense disambiguation in our definition selection step and explore a fully automated word clustering algorithm that does not depend on WordNet relations. An application of adjective intensity ordering could extract gradable adjectives used in analyzing reviews to compare on a particular scale, such as happiness. Our code and data set is available at: github.com/CRESST-NLP/AdjectiveIntensityOrdering

The authors would like to thank Joanne Michiuye and Alana Kennedy for the help with rating adjectives.

Appendix A
Happiness Rankings

Table III: 1 is the least happy and 137 is the most happy. The number after the period is the WordNet dictionary sense.

TABLE III. Happiness Rankings

Happiness	LS	H	Happiness	LS	H
unsatisfied.02	1	37	rhapsodic	70	132
suffering.02	2	2	riant	71	81
discontent	3	66	light-hearted	72	86
restless	4	50	lighthearted	73	84
depressed.03	5	42	buoyant.02	74	105
unhappy	6	51	jocund	75	115
rebellious.02	7	46	lightsome.02	76	85
ill-affected	8	56	blithesome	77	117
wretched.03	9	1	blessed	78	75
disaffected	10	67	enraptured	79	127
discontented	11	59	euphoric	80	137
malcontent	12	48	exulting	81	129
miserable	13	9	gladdened	82	93
uncheerful	14	60	gleeful	83	119
disconsolate.02	15	40	joyful	84	101
disgruntled	16	64	jubilant	85	125
down.09	17	41	laughing	86	99
unhappy.02	18	49	prideful.02	87	80
displeased	19	63	rejoicing	88	128
downhearted	20	30	smiling	89	83
dissatisfied	21	61	triumphal.02	90	134
depressing	22	53	triumphant	91	135
woebegone.02	23	7	twinkly	92	76
down in the mouth	24	47	joyous	93	100
rapturous	25	131	uplifted	94	78
mirthful	26	104	contented	95	68
low.10	27	28	self-complacent	96	70
joyless	28	62	content	97	69
somber.02	29	22	satisfied	98	71
sombre.02	30	21	gladsome	99	90
heavyhearted	31	23	gay.02	100	116
low-spirited	32	29	gay.05	101	114
blue.03	33	45	halcyon.02	102	88
dejected	34	52	high.05	103	97
chapfallen	35	58	smug	104	74
gloomy.02	36	36	pleased	105	79
gloomy.03	37	35	glad.01	106	96
glum	38	33	bright.09	107	102
sorrowful	39	8	glad.04	108	92
heartsick	40	20	beaming	109	120
dismal	41	43	cheerful	110	122
grim.05	42	26	cheery	111	103
grim.06	43	25	chipper	112	110
sepulchral.03	44	31	chirpy.02	113	107
sorrowing	45	6	debonair.02	114	98
lachrymose	46	16	debonaire.02	115	95
cheerless	47	65	festal	116	111
crestfallen	48	55	festive	117	108
dolorous	49	34	gay	118	118
dolourous	50	32	happy	119	91
dreary.02	51	39	in high spirits	120	94
melancholic	52	17	jaunty.02	121	87
melancholy	53	15	jolly	122	113
melancholy.02	54	14	jovial	123	112
sad	55	27	merry.01	124	109
sad.02	56	19	merry.02	125	106
sorry.04	57	13	perky	126	82
wistful	58	54	sunny	127	77
grief-stricken	59	24	self-satisfied.02	128	72
tragic	60	3	self-satisfied	129	73
woeful	61	5	beatific	130	123
elegiac.02	62	44	elated	131	126
lugubrious	63	18	ecstatic	132	136
plaintive	64	12	exultant	133	130
mournful	65	11	exhilarated	134	124
doleful	66	38	golden.02	135	89
funereal	67	57	blissful	136	121
wailful	68	4	overjoyed	137	133
mourning	69	10			

References

[1] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–1543.

[2] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995. [Online]. Available: http://doi.acm.org/10.1145/219717.219748

[3] V. Sheinman and T. Tokunaga, "Adjscales: Visualizing differences between adjectives for language learners," *IEICE Transactions on Information and Systems*, vol. E92.D, no. 8, pp. 1542–1550, 2009.

[4] V. Sheinman, C. Fellbaum, I. Julien, P. Schulam, and T. Tokunaga, "Large, huge or gigantic? identifying and encoding intensity relations among adjectives in wordnet," *Language Resources and Evaluation*, vol. 47, no. 3, pp. 797–816, Sep 2013. [Online]. Available: https://doi.org/10.1007/s10579-012-9212-1

[5] H. Wang, X. H. Ji, and J. Ma, "Automatically collect the adjectives and calculate their sentiment intensity in the sentiment analysis," in *Mechanical Components and Control Engineering III*, ser. Applied Mechanics and Materials, vol. 668. Trans Tech Publications, 11 2014, pp. 1138–1142.

[6] C. Paradis, *Degree modifiers of adjectives in spoken British English*, ser. Lund Studies in English. Lund University Press, 1997, vol. 92.

[7] J. Ruppenhofer, J. Brandes, P. Steiner, and M. Wiegand, "Ordering adverbs by their scaling effect on adjective intensity," in *RANLP*, 2015.

[8] P. Bora, *A Real Dictionary Module for Python*, 2014, python Package PyDictionary 1.3.4. [Online]. Available: https://pypi.python.org/pypi/PyDictionary/1.3.4

[9] M. Schulze, "A new monotonic, clone-independent, reversal symmetric, and condorcet-consistent single-winner election method," *Social Choice and Welfare*, vol. 36, no. 2, pp. 267–303, Feb 2011. [Online]. Available: https://doi.org/10.1007/s00355-010-0475-4

[10] A. F. Hayes and K. Krippendorff, "Answering the call for a standard reliability measure for coding data," *Communication Methods and Measures*, vol. 1, no. 1, pp. 77–89, 2007. [Online]. Available: http://dx.doi.org/10.1080/19312450709336664

[11] M. Gamer, J. Lemon, I. Fellows, and P. Singh, *irr: Various Coefficients of Interrater Reliability and Agreement*, 2012, r package version 0.84. [Online]. Available: https://CRAN.R-project.org/package=irr